Vehicle Loan Default Prediction

by:

Abdulrahman Alrubaiya

Mohammed Alghamdi

Content

- Introduction
- Tools
- Baseline
- Feature selection and modeling
- Lessons learned

Introduction

- Financial institutes are suffering from losses
- Striving to achieve better credit scoring model
- Predict car loan defaults

Tools used and frameworks

- Python
  - Pandas
  - Numpy
  - Matplotlib, Seaborn
  - Scikit-learn
  - Flask
- Tableau 2021
- Heroku For deployment

Data origins

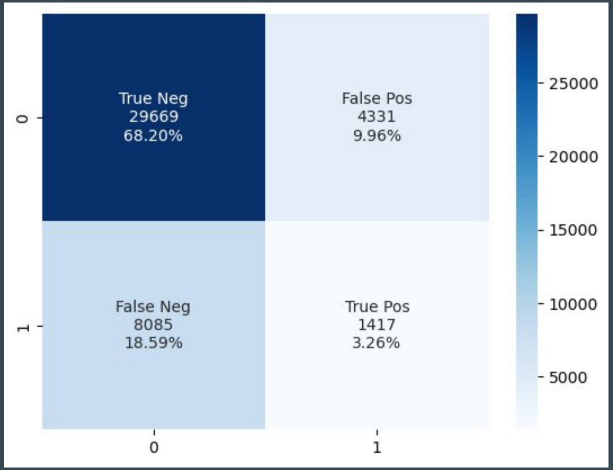
- Obtained from Kaggle
- Part of a FinHackathon competition
- 270k+ rows, 41 columns

Baseline

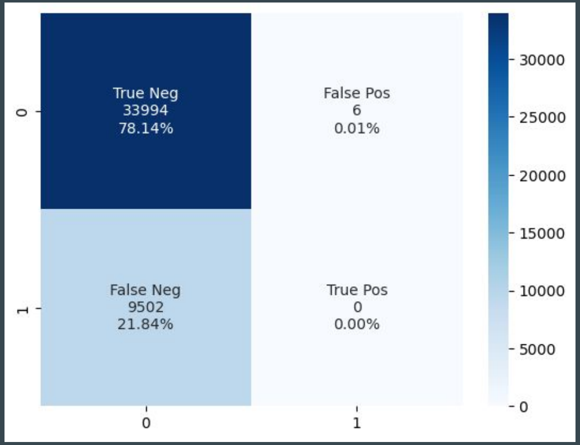
- KNN train score: 0.8322
- KNN validation score:0.7145

- Logistic train score: 0.7789
- Logistic validation score: 0.7814

KNN confusion matrix



Logistic confusion matrix



Data preprocessing and feature engineering

- Date conversion to days
- Remove outliers (Upper limit, lower limit, intuitive judgement)
- Binning for risk score (Categories)
- Create new column out of two features (Loan to asset ratio)

Feature selection

- Approach was based on:
    - Running feature selection techniques:
        - Lasso model
        - SelectFromModel (sklearn meta-transformer)
        - ~~Linear discriminant analysis~~
    - Multicollinearity (2 removed in this process)
    - Variance inflation factor (3 removed in this process)
    - Business domain knowledge
    - 14 features left

$$VIF = \frac{1}{1-R^2}$$

Modeling

- Imbalanced (class 1 is ~77% and class 2 is ~23%) ~1:3 Ratio
  - Oversampling (Using SMOTE)
- GridsearchCV
- Models:
  - Logistic Regression
  - Random Forest
  - KNN
  - Voting Classifier
  - XGBoost

Logistics Regression

	Accuracy	Precision	Recall	F1
Train	0.64	0.64	0.64	0.63
Validation	0.46	0.69	0.46	0.50

I am 69% correct in predicting class 1
I correctly classified 46% of class 1

Adjusting threshold

threshold >= 0.324

	Accuracy	Precision	Recall	F1
Validation	0.30	0.71	0.30	0.25

Not what we want, discard changes

Random Forest

	Accuracy	Precision	Recall	F1
Train	0.78	0.80	0.79	0.78
Validation	0.61	0.69	0.58	0.61

KNN

K=22

	Accuracy	Precision	Recall	F1
Train	0.69	0.70	0.69	0.69
Validation	0.58	0.55	0.58	0.55

XGBoost

learning_rate=0.345

	Accuracy	Precision	Recall	F1
Train	0.93	0.93	0.93	0.93
Validation	0.63	0.68	0.63	0.65

Voting Classifier

[Logistics Regression, Random Forest, KNN, XGBoost] ‘hard’

	Accuracy	Precision	Recall	F1
Train	0.77	0.77	0.77	0.77
Validation	0.51	0.69	0.51	0.55

Choice of model

	Accuracy	Precision	Recall	F1
Train	0.93	0.93	0.93	0.93
Test	0.67	0.67	0.67	0.67

XGBoost!

Lessons learned

- Plan your approach
- Allocate more time for modeling
- Trained models sharing using joblib
- Sklearn library doesn’t benefit from GPU computing
  - n_jobs=-1

Thank you